X-Y Controller

Software Functional Requirements

By **Gary Twinn**

Contents

List of Figures

List of Tables

Overview

The X-Y controller software consists of a python (Python Software Foundation, 2020) application to control the 2 stepper motors on the X-Y Stage of the Helium line.

The X-Y table consists of 2 Bipolar stepper motors to drive the table in two dimensions. A stepper motor consists of a pair of windings A-AA and B-BB surrounding a rotor containing permanent magnets (Figure 1)

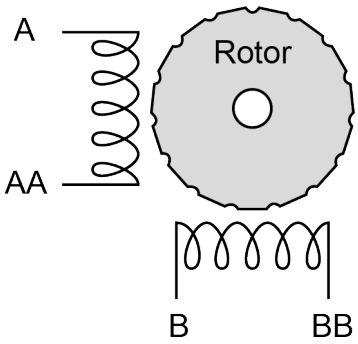

Figure 1: Bipolar Stepper Motor

The stepper motor is operated by applying a voltage across the A and B windings which causes the permanent magnets on the rotor to either attract or repel. The voltage is applied a with the positive to A and B and the negative to AA and BB. To move the rotor one winding will have the voltage reversed in turn. At each transition where the voltage is reversed on a winding the rotor will rotate 1.8° clockwise. By reversing the sequence the rotor will rotate anti-clockwise. By grounding both connections during of one winding between the reversals will cause the rotor to take a half step. Adding a delay between transitions will set a speed for the motor, a fast speed would have 25ms between transitions whereas a slower speed is used for accurately locating the laser over the sample.

	A	AA	B	BB	
0	+	-	+	-	Initial State
1	+	-	-	+	B reversal
2	-	+	-	+	A reversal
3	-	+	+	-	B revert
4	+	-	+	-	A revert

Table 1: Full step sequence for bipolar stepper motor

	A	AA	B	BB	
0	+	-	+	-	Initial State
1	+	-	-	-	½ step ground B
2	+	-	-	+	B reversal
3	-	-	-	+	½ step ground A
4	-	+	-	+	A reversal
5	-	+	-	-	½ step ground B
6	-	+	+	-	B revert
7	-	-	+	-	½ step ground A
8	+	-	+	-	A revert

Table 2: Half step sequence for bipolar stepper motor

There are two 2KΩ linear displacement sensors with a 50mm travel mounted on the x-y stage, one in each direction. By applying a voltage to each end of the sensor, the location of the wiper along the sensor can be determined. The travel along the planchet is 10.5mm in each of the X directions (from the origin) and 17.5mm from the origin in each of the Y directions.

The 2KΩ displacement sensor will have a reference voltage of +5.08 volts applied to one static connection, the other static connection will be held to 0V. That will give a voltage gradient along the 50mm of 0.1V per mm.

A joystick will be required for manual control of the X-Y table.

The computer required to control the valves will be a Raspberry Pi 4 (Raspberry PI Foundation, 2020), it will require a connector to break out the required 12 GPIO pins and connect to the driver boards.

Each stepper motor will be connected to 4 dedicated GPIO lines on the raspberry Pi computer. The valves run on 5V DC @ 1 A, so a driver circuit will be required to step up the voltage and current from the Pi. The linear

The Raspberry Pi computer will be controlled via a RESTful API listening on port 80 for a valid json message.

Input Messages

Json messages in the following formats will be accepted:

Move in X direction to a location (voltage). Where n is a floating-point number between -2.5 and +2.5. Once started the stepper will move at a relative high speed (30 steps per second) until the position voltage is within 0.2V (~ 2mm) at which point it will slow to a rate of 10 steps a second until the position voltage is within 0.1V (~ 1mm). At 0.1V the rate will slow again to 3 steps per second until the desired position has just passed, at that point the stepper will reverse for 1 step to position as the table as accurately as possible with the stepper motors attached to the x-y stage.

```
{
    "item": "xmoveto",
    "command": n
}
```

Move in X direction nn steps. Where nn is a positive or negative integer depending on the direction or motor rotation. Sending a value of 0 will cause the stepper motor to halt immediately and disengage power to the x direction stepper motor.

```
{
    "item": "xmove",
    "command": nn
}
```

Move in Y direction to a location (voltage). Where n is a floating-point number between -2.5 and +2.5.

```
{
    "item": "ymoveto",
    "command": n
}
```

Move in Y direction nn steps. Where nn is a positive or negative integer depending on the direction or motor rotation. Sending a value of 0 will cause the stepper motor to halt immediately and disengage power to the y direction stepper motor.

```
{
    "item": "ymove",
    "command": nn
}
```

Output Messages

Following any valid command, the following query is returned: (ss = open or closed)

```
{
    "xpos": xnn,
    "ypos": ynn
}
```

Where xnn and ynn are floating point numbers denoting the output voltages of each displacement sensor.

Control to GPIO assignment

Control	Designation	Connector		Designation	Control
	3v3	1	2	**5v**	
i2c Data	GPIO 02	3	4	**5v**	
i2c Clock	GPIO 03	5	6	**GND**	
GP clock	GPIO 04	7	8	GPIO 14	
	GND	9	10	GPIO 15	
X Stepper A	**GPIO 17**	11	12	**GPIO 18**	Y Stepper A
X Stepper AA	**GPIO 27**	13	14	**GND**	
X Stepper B	**GPIO 22**	15	16	**GPIO 23**	Y Stepper AA
	3v3	17	18	**GPIO 24**	Y Stepper B
	GPIO 10	19	20	**GND**	
Y Stepper BB	**GPIO 09**	21	22	GPIO 25	
Joystick X+	**GPIO 11**	23	24	GPIO 08	
	GND	25	26	GPIO 07	
	GPIO 00	27	28	GPIO 01	
	GPIO 05	29	30	**GND**	
	GPIO 06	31	32	**GPIO 12**	Ready LED
X Stepper BB	**GPIO 13**	33	34	**GND**	
	GPIO 19	35	36	**GPIO 16**	Joystick X-
Joystick Speed	**GPIO 26**	37	38	**GPIO 20**	Joystick Y+
	GND	39	40	**GPIO 21**	Joystick Y-

Table 3: Control to GPIO Assignments

The above assignments are based on a raspberry Pi 4B. GPIO channels in green are available and remain at 0v during the Pi boot up.

Once the boot sequence has completed and the software has started the final command on the initialising function will be to light the “Ready LED” to give a visual indication the raspberry PI has booted and is ready to accept commands on the REST api.

The Joystick connections will need to be set to inputs with an internal pull-up resistor, thus requiring a connection to GND in order to trigger movement.

Driver circuit and power supply

The Pi 4B uses a 5v power supply provided via a USB-C connector. A 5V power supply rated at 3A will be required.

The GPIO outputs are 3.3v at 50mA, the stepper motors require 5V DC @1A to rotate but the current could increase above 1A under a heavy load. In order to drive the 4 windings in the stepper motors a driver circuit will be required to step up the voltage and current (Figure 2). A power supply with a continuous rating of 6.0A will be required to ensure reliability. To protect the Pi Zero from any voltage spikes that may be generated by the steppers, or a faulty component on the hight voltage/high current side of the driver, an opto-isolator will be required to provide full electrical isolation between the Pi and the driver circuit.

The stepper motors will be driven via an integrated circuit containing four Metal Oxide Field Effect Transistors (MOSFET). MOSFETs have a very low forward voltage once switched on and have a high-power capacity. As the stepper windings are inductive loads, when the power is removed from the windings there is a chance voltage spikes will be induced into the circuit so a Schottky diode will protect the MOSFET and opto-isolator.

An analogue to digital convertor will be required to read the voltage from the position sensors and turn it into a signal the Raspberry Pi computer can read. The ADC chosen was from Able Electronics (Able Electronics UK 2020) and a resolution of 12 bits was chosen giving the controller the ability to detect 4096 voltage steps. Given that a 50mm travel on the position sensor would give a voltage reading between 0V and 5V the voltage to distance conversion is 0.1V / mm. Dividing the 5V supply by the step resolution steps gives 0.0012V per step or a resolution of 0.012 mm.

The driver circuits will need to be housed in an enclosure that has external connectors for the mains supply, ethernet cable and connectors for the x and y steppers, x and y position sensors and joystick cables. It must have ventilation holes to prevent the Raspberry Pi’s, power supplies or driver MOS-FET transistors overheating. Indicator lights should show the status of the 5v PSU for the Raspberry Pi, the 5V Stepper PSU and the ready light to show the software has started.

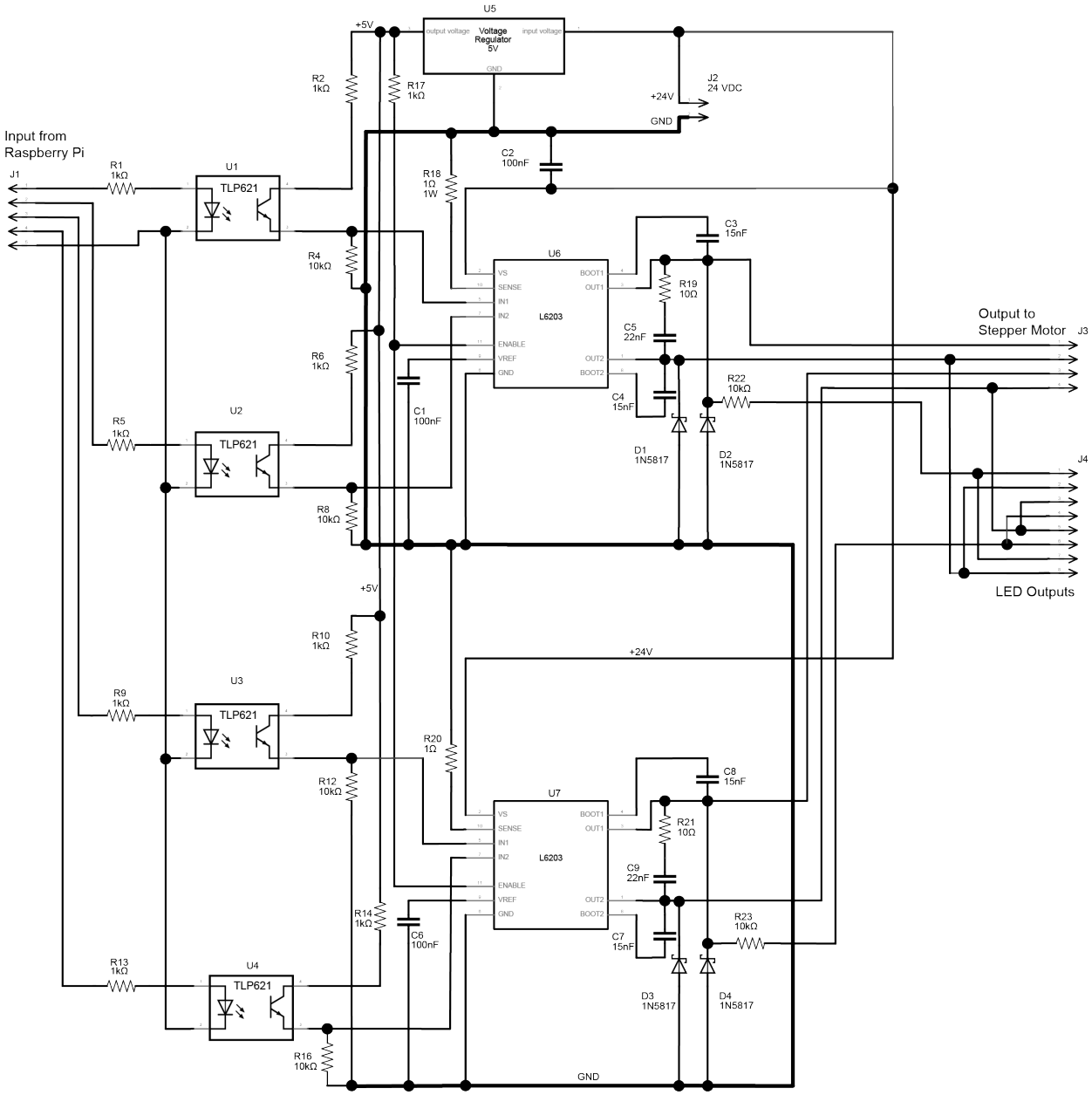

Figure 2: Schematic for a valve / stepper driver board

Web interface

As well as accessing the status of the valves via the RESTful API a read-only web interface that can be accessed directly from a browser is will be available to view the valve status, application and web server logs.

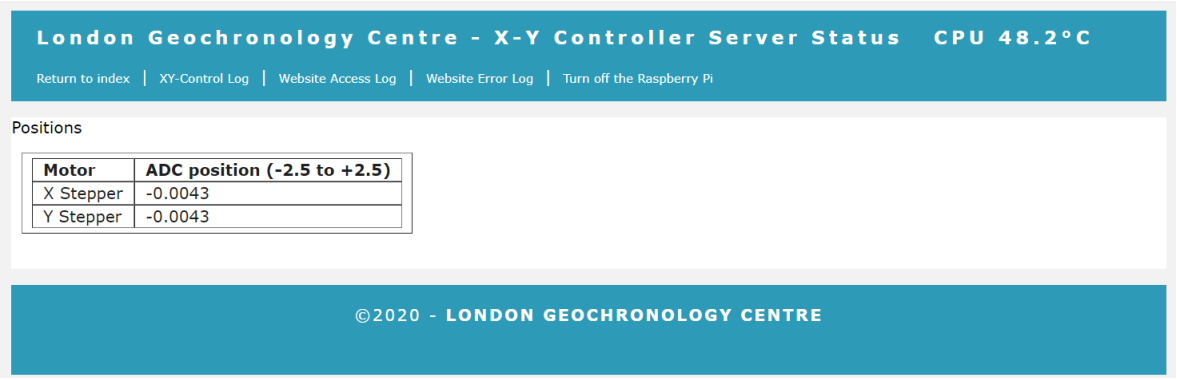

Figure 3: Web status page

Interfaces

The connection from the back of the housing X and Y steppers, X and Y position sensors and Joysticks will be a standard 9pin D Plugs

Stepper drive cable specification

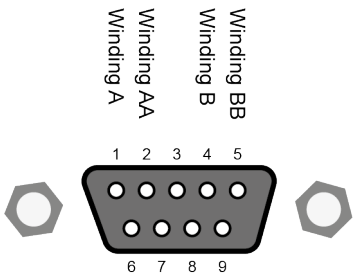

Figure 4: Stepper motor D plug assignments

The stepper drive cable consists of a male 9 pin D-plug for the controller connection, a 6 way cable with a minimum of a 1 amp capacity and a female 9 pin D-Plug for the stepper motor connector. Two cables are required, one for each stepper

Position sensor cable specification

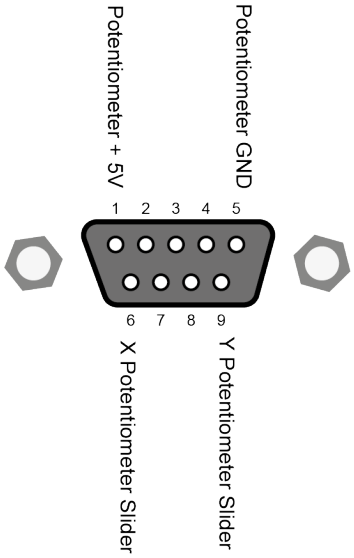

Figure 5: Position sensor D plug assignments

The position sensor cable consists of a male 9 pin D-plug for the controller connection, a 4 way shielded cable and a 4 pin DIN connector for the X-Y table connection.

Joystick Connection Specification:

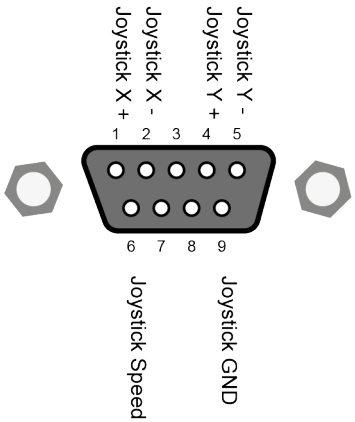

Figure 6: Joystick D plug assignments

Operating System Installation

Operating system

Use BalenaEtcher to install the latest version of the buster-lite operating system onto a 32Gb MicroMMC card.

Run the `sudo raspi-config` command to:

- enable ssh
- disable Serial
- enable 1^2C bus
- set the password on the pi user account to something other than “raspberry”
- change the GPU memory to 16Mb

Software installation

Run Operating System Updates

Run `sudo apt update`
Run `sudo apt upgrade`

Install SMBUS

Run `sudo apt-get install python3-smbus`
Run `sudo apt-get install i2c-tools`

Check that the ADC is connected and working:
Run `sudo i2cdetect -y 1`
You should get an output like:

```
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 69 -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

Install PIP3 for Python 3.x installation

Run `sudo apt install python3-pip`

Install Flask installation

Run `sudo pip3 install flask`

Install Raspberry Pi GPIO Driver

Run `sudo pip3 install rpi.gpio`

Install SMBUS

Run `sudo pip3 install smbus2`

Download the XY-Controller application

Download the files from https://github.com/westerlymerlin/UCL-RPi-XY-Controller.git

Copy the python files to the Pi

- Copy the files from GitHub to `/home/pi/`
- Copy the folder templates to `/home/pi/templates`
- Copy the folder static to `/home/pi/static`
- Create a folder `/home/pi/database`
- Create a folder `/home/pi/logs`
- `Copy the files /raspberry-pi/home/pi to /home/pi`

Nginx installation

Run `sudo apt install nginx`

Copy the Github file
`\raspberry-pi\home\pi\etc\nginx\nginx.conf`
to
`/etc/nginx/nginx.conf`

Copy the GitHub file
`\raspberry-pi\home\pi\etc\nginx\sites-available\icpmsdata`
to
`/etc/nginx/sites-available/icpmsdata`

Change directory to `/etc/nginx/sites-enabled/`
Run `sudo rm default`
Run `sudo ln -s /etc/nginx/sites-available/icpmsdata`

Gunicorn for Python 3.x installation

Run `sudo apt install gunicorn3`

Copy the GitHub file
`\raspberry-pi\home\pi\etc\systemd\system\gunicorn.service`
to
`/etc/system/system/gunicorn.service`

Run `sudo systemctl enable gunicorn`

Run `sudo systemctl start gunicorn`

If flask is installed, the python files are in the `/home/pi` directory, gunicorn3 is installed and configured and nginx is installed and configured the web service should be running and the site will be accessible on http:// (ip address of the server)

References

Able Electronics UK (2020) *ADC Pi*. Available online: https://www.abelectronics.co.uk/p/69/adc-pi-raspberry-pi-analogue-to-digital-converter [Accessed July 2020].

Python Software Foundation (2020) *Python 3 Programming Language*. Online. Available online: https://www.python.org [Accessed July 2020].

Raspberry PI Foundation (2020) *Raspberry PI Model 4B Reference*. Available online: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/ [Accessed July 2020].